Tabu pretraga za Vehicle Routing Problem

Projekat iz Računarske inteligencije

Matematički fakultet

Univerzitet u Beogradu

Srđan Katančević

mi18268@alas.matf.bg.ac.rs

Sadržaj

1 Opis problema

Vehicle routing problem (Problem rutiranja vozila) je problem određivanja optimalnih ruta kojima ce se vozila kretati kako bi dostavila korisnicima određenu količinu robe. Vozila kreću iz polaznog skladišta, razvoze robu do korisnika i vraćaju se u polazno skladište. Samo jedno vozilo je dozvoljeno da obiđe jednu mušteriju. Rešenje problema je određivanje ruta koje zadovoljavaju određene uslove i za koje važi da je trošak prevoza minimalan.

Postoje razne varijante VRP problema:

- Capacitated VRP (CVRP)-vozila imaju limit robe koje mogu da prevoze
- VRP with profits (VRPP)-cilj nije obilazak svih mušterija već maksimizacija profita
- VRP with Time Windows(VRPTW)- za svaku lokaciju postoji vremenski rok za dostavu robe
- ...

U nastavku teksta ćemo razmatrati CVRP.

2 Tabu pretraga uopšteno

Tabu pretraga je metaheuristika koja spada u algoritme lokalne pretrage i koristi se za rešavanje problema kombinatorne optimizacije. Cilj ove vrste pretrage je da izbegne zaglavljivanje u lokalnim minimumima, što se postiže korišćenjem zabranjenih stanja (tabua). Održava se tzv. tabu lista, tj. lista stanja u kojima je pretraga već bila i u koja se više neće vraćati. Dozvoljavaju se potezi koji pogoršavaju vrednost funkcije, pod uslovom da nema boljih dozvoljenih poteza. Ova dva pristupa omogućavaju da se napusti lokalni minimum. Detaljnije o tabu pretrazi se može naći u ovoj knjizi [1].

3 Definicija problema

Neka su dati M,P,v,k gde je M matrica dimenzija n puta n gde je n broj mušterija i element matrice $e_{i,j}$ predstavlja rastojanje između mušterije i i mušterije j, P niz elemenata p_i gde je p_i potraznja mušterije i, v broj vozila i k kapacitet vozila. Rešenje mora ispuniti sledeće uslove:

1. Svaka ruta počinje i završava se u skladištu
2. Svaka mušterija je posećena tačno jednom od tačno jednog vozila
3. Ukupna potražnja mušterija koja je dodeljena jednoj ruti ne sme da premašuje k
4. Ukupna cena prevoza je minimalna

4 Pohlepni algoritam

Pohlepni algoritam se vrti u petlji sve dok postoji neka mušterija koja nije posećena, pronalazi najbližu mušteriju u odnosu na trenutno vozilo i ako potražnja te mušterije može da stane u trenutno vozilo, onda algoritam upisuje mušteriju u rutu tog vozila, označava je kao posečenu i ažurira nosivost tog vozila. Ako više ne postoji mušterija čija potražnja može da stane u trenutno vozilo, trenutno vozilo se vraća u skladište i algoritam prelazi na sledeće vozilo (sva vozila kreću iz skladišta). Algoritam se zaustavlja kada je obišao sve mušterije ili kada nema više slobodnih vozila.

GreedySolution

```
while(true):
for v in vozila:
nadji najblizi cvor c u odnosu na trenutnu poziciju v i
        c.potraznja+v.opterecenje<v.kapacitet;
                if  takav cvor c ne postoji :
                        dodaj skladiste na kraj rute vozila v;
                        continue;
                else:
                dodaj cvor c na kraj rute vozila v i povecaj opterecenje od v za
                        potraznju od c;
```

if nema vise neposecenih cvorova:
break;

---

5 Tabu Pretraga

Tabu pretraga započinje kao obična lokalna pretraga, iterativno prolazi od jednog rešenja do drugog sve dok nije ispunjen neki kriterijum zaustavljanja i koristi neke strategije kako se ne bi zaglavila u lokalnom optimumu.

5.1.1 Prostor pretrage

Prostor pretrage sadrži sva moguća rešenja koja su relevantna za dati problem i koja će biti uzeta u razmatranje tokom pretrage. Za naš problem, prostor pretrage čine sve moguće rute koje ispunjavaju sva ograničenja navedena u definiciji problema.

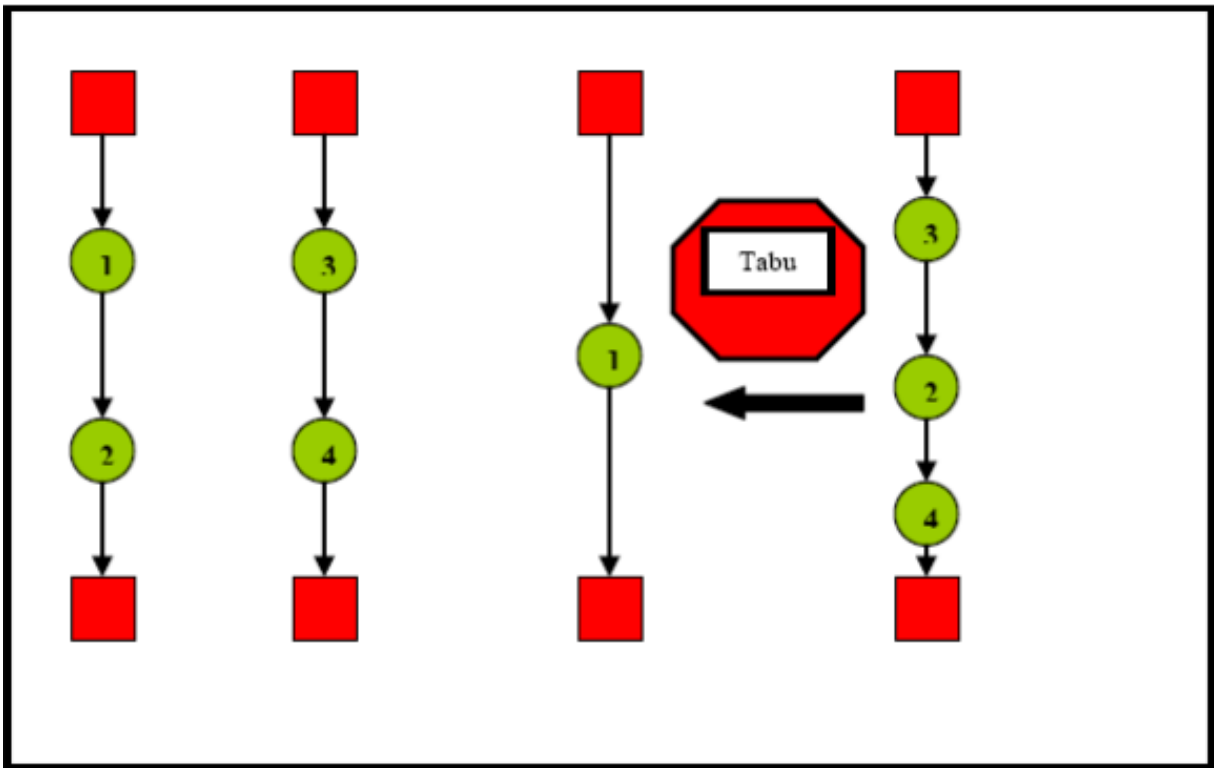
5.1.2 Okoline

Okolina nekog rešenja x se dobija tako što se vrši izmena tog rešenja u svakoj iteraciji i tako se dobija okolina N(x). Postoje dve vrste okolina: jednostavna i kompleksna. Jednostavna okolina se dobija primenom jednostavnih izmena na trenutno rešenje. Primer jednostavne izmene bi bio pomeranje jedne mušterije iz neke rute u neku drugu rutu ili na neko drugo mesto u istoj ruti.

5.1.3 Tabu liste

“Tabu” potez je fundamentalna osobina Tabu pretrage. “Tabu” potezi se koriste za izbegavanje kruženja kada se pretraga udaljava od lokalnog optimuma koristeći lošije poteze. Naprimer, ako je mušterija 1 iz rute 1

pomerena u rutu 2, onda se potez pomeranja mušterije 1 iz rute 2 u rutu 1 može deklarisati kao “Tabu” na neki broj iteracija.



Slika 1: Primer tabu poteza

Tabu tenure je broj iteracija koliko se neki potez smatra “Tabu” potezom. Tabu potezi se čuvaju u strukturi podataka koja se zove tabu lista. Tabu lista može da se realizuje na više načina:

1. Lista čuva potpuna rešenja
2. Lista čuva samo poslednjih par poteza kojima smo došli do trenutnog rešenja

Međutim tabu liste imaje neke nedostatke, one mogu da zabrane neke poteze iako ti potezi nebi stvarali opasnost od vraćanja unazad i sveukupno tabu liste mogu da dovedu do stagnacije procesa pretrage.

5.1.4 Kriterijumi zaustavljanja

U nekom trenutku proces pretrage mora biti zaustavljen. U algoritmima tabu pretrage postoje različiti kriterijumi zaustavljanja:

1. Izvršen je unapred određen broj iteracija
2. Određeni broj iteracija nema poboljšanja rešenja
3. Kada trenutno najbolje rešenja ispunjava zahteve

5.2 Generalna formulacija algoritma Tabu pretrage

Korak 1: Izaberi inicijalno rešenje i u S

Postavi $i^*=i$ $k=0$

Korak 2: $k = k + 1$ i generiši podskup rešenja V* iz N(i, k)

Korak 3: Izaberi najbolje rešenje j iz V* i postavi $i = j$

Korak 4: Ako je $f(i)<f(i^*)$ onda postavi $i^*=i$

Korak 5: Ažuriraj tabu listu

Korak 6: Ako je ispunjen kriterijum zaustavljanja onda stani, inače se vrati na korak 2

Notacije:

i, j - indeksi rešenja

k - indeks iteracije

V* - podskup rešenja

N(i, k) - okolina rešenja i u iteraciji k

$f(i)$ - vrednost funkcije cilja za rešenje i

---

6 Rešavanje VRP korišćenjem Tabu pretrage

6.1 Kreiranje inicijalnog rešenja

Kao inicijalno rešenje ćemo koristiti rešenje koje je dobijeno pohlepnim algoritmom.

6.2 Kreiranje okoline rešenja (neighbourhoods)

Okoline se kreiraju tako što par mušterija unutar jedne rute zameni mesta ili tako što se mušterija iz rute jednog vozila prebaci u rutu drugog vozila.

6.3 Kriterijum zaustavljanja

Kriterijum zaustavljanja će biti unapred odredjen broj iteracija izvršavanja programa.

6.4 Tabu lista

Začuvanje tabu poteza ćemo koristiti matricu n puta n gde element matrice eij predstavlja koliko iteracija je potez zamene mušterije i mušterijom j tabu tj. zabranjen. Ako je vrednost elementa 0 onda taj potez nije tabu. Svaki put kada se izvrši neki potez on postaje tabu na određeni broj iteracija, taj broj iteracija je unapred određen i u našem algoritmu će on imati vrednost 10.

6.5 TabuSolution

Kao inicijalno rešenje algoritma uzimamo rešenje pohlepnog algoritma. Sve dok nije ispunjen kriterijum zaustavljanja za trenutno rešenje tražimo rešenje iz njegove okoline takvo da je ukupna cena prevoza iz rešenja iz okoline manja od trenutnog i manja od svi ostalih rešenja iz okoline. Takvo rešenje pronalazimo tako što prolazimo kroz listu vozila, za svako vozilo v1 iz te liste prolazimo kroz listu čvorova u njegovoj ruti i za čvor c1 i tražimo čvor c2 iz rute tog ili nekog drugog vozila v2 takav da se ubacivanjem čvora c1 iz vozila v1 posle čvora c2 u vozilu v2 (vozila v1 i v2 mogu biti isto vozilo) dobija novo rešenje čija je ukupna cena manja od trenutnog rešenja i zadovoljen je uslov nosivosti vozila (suma potražnji mušterija unutar rute nekog vozila ne sme da bude veća od nosivosti vozila) i nijedan od poteza nije tabu. Kada smo obišli sva vozila vršimo zamenu trenutnog

rešenja sa najboljim rešenjem, ažuriramo tabu matricu tako što povećamo za 10 vrednost elemenata koji predstavljaju poteze koje smo koristili za dobijanje ovog novog rešenja. Na kraju iteracije najbolje rešenje postaje trenutno i smanjujemo trajanje svih tabu poteza za 1.

7 Testiranje

Test primeri nad kojima ćemo testirati naš program su preuzeti sa [3] i oni su u TSPLIB formatu. Broj iteracija za svaki test je iznosio 10 000.

Ime test primer	Dimenzija problema	Pohlepni algoritam	Tabu pretraga	Optimalno rešenje
A-n32-k5.vrp	32	1145	785	784
A-n33-k5.vrp	33	977	666	661
A-n53-k7.vrp	53	1444	1137	1010
A-n62-k8.vrp	62	1775	1345	1290
A-n80-k10.vrp	80	2348	1893	1764
M-200-k17.vrp	200	1989	1483	1335

8 Literatura

[1] Fred Glover, Manuel Laguna - Tabu search

[2] Caroline Nadal - Using Tabu Search Heuristic in Solving the Vehicle Routing Problem with Time Windows

[3] http://vrp.atd-lab.inf.puc-rio.br/index.php/en/